

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-053235

(43)Date of publication of application : 26.02.1999

(51)Int.Cl.

G06F 12/00
G06F 3/06

(21)Application number : 09-214656

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 08.08.1997

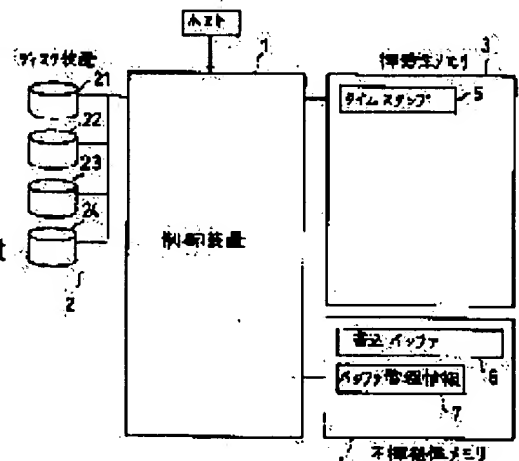
(72)Inventor : SEKIDO KAZUNORI

(54) DATA UPDATING METHOD OF DISK STORAGE DEVICE AND DISK STORAGE CONTROL SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data updating method for a disk storage device which does not need an indirect map in principle and is inexpensive and fast and also to construct a disk storage control system which realizes the method.

SOLUTION: This disk storage device which consists of N disk drives is provided with a write buffer 6 which has a capacity that corresponds to $N \times K$ (integer) logical blocks and accumulates a logical block of data to be updated in the write buffer. A controller 1 delays data updating of the logical block until the accumulated logical blocks reach $N \times K - 1$, generates a logical address tag block that consists of logical addresses for each logical block accumulated in the write buffer, and continuously and sequentially writes $N \times K$ logical blocks which add the logical address tag blocks to $N \times K - 1$ logical blocks in an empty that is different from an area that holds data to be updated on N disk devices.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-53235

(43) 公開日 平成11年(1999) 2月26日

(51) Int.Cl.⁶

G 0 6 F 12/00
3/06

識別記号

5 1 4
5 4 0

F I

G 0 6 F 12/00
3/06

5 1 4 A
5 4 0

審査請求 未請求 請求項の数26 O L (全 13 頁)

(21) 出願番号 特願平9-214656

(22) 出願日 平成9年(1997) 8月8日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 関戸 一紀

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

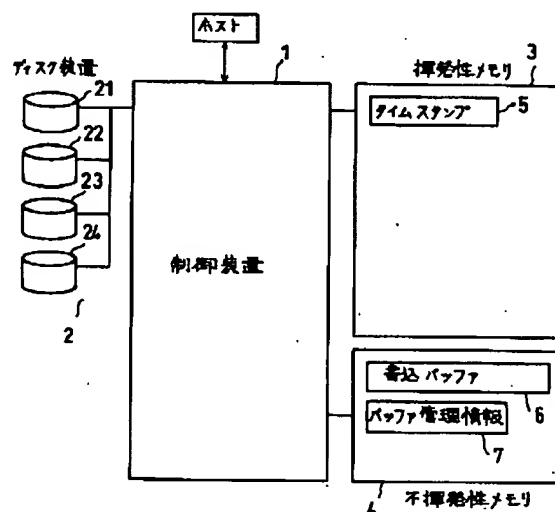
(74) 代理人 弁理士 大胡 典夫 (外1名)

(54) 【発明の名称】 ディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システム

(57) 【要約】

【課題】 本発明は、間接マップを原理的に必要としない、安価で高速なディスク記憶装置のデータ更新方法を提案するものであり、併せて同方法を実現するディスク記憶制御システムを構築することを課題とする。

【解決手段】 N台のディスク装置から構成されるディスク記憶装置において、 $N \times K$ (整数) 個の論理ブロックに相当する容量を持つ書込みバッファ6を備え、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、制御装置1は、その蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックのデータ更新を遅延させ、書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、 $N \times K - 1$ 個の論理ブロックに前記論理アドレスタグブロックを加えた $N \times K$ 個の論理ブロックを、N台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む。



【特許請求の範囲】

【請求項1】 N台のディスク装置から構成されるディスク記憶装置において、 $N \times K$ （整数）個の論理ブロックに相当する容量を持つ書込みバッファを有し、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、 $N \times K - 1$ 個の論理ブロックに前記論理アドレスタグブロックを加えた $N \times K$ 個の論理ブロックを、N台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込むことを特徴とするディスク記憶装置のデータ更新方法。

【請求項2】 上記書込みは、複数のディスク装置に上記物理ブロックが並行して書込まれることを特徴とする請求項1記載のディスク記憶装置のデータ更新方法。

【請求項3】 上記論理アドレスタグブロックに書き込み順序を判定するために用いられるタイムスタンプを付加することを特徴とする請求項1記載のディスク記憶装置のデータ更新方法。

【請求項4】 ディスク装置に記録された論理アドレスタグブロックを検査し、各論理アドレスに対応するディスク装置上の位置を見つけることを特徴とする請求項1記載のディスク記憶装置のデータ更新方法。

【請求項5】 上記の検査において、同じ論理アドレスを含むストライプ（ $N \times K$ 個の論理ブロック）が複数ある場合、タイムスタンプが最新のストライプのブロックを有効ブロックとし、他の同論理アドレスを持つブロックを無効ブロックと判定することを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項6】 上記の検査において、最大のタイムスタンプ値を見つけ、次の書込みで付加するタイムスタンプを再生することを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項7】 上記の検査において、最小のタイムスタンプ値を見つけ、書込み順序の判定基準となるタイムスタンプ値を求めることを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項8】 上記ディスク装置上に論理ブロックを連続して書込めるような空領域を作るため、複数ストライプを読み出して有効ブロックだけを前記書込みバッファに移し、対応する論理タグブロック内の論理アドレスから新しい論理アドレスタグブロックを生成し、書込みバッファの有効データと生成された論理アドレスタグから構成されるストライプを、読み出したストライプを保持していた領域とは別の空領域に順次書込むことを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項9】 上記新しい論理アドレスタグブロックを

生成するとき、有効ブロック数が $N \times K - 1$ 個に満たない場合は、新しい論理アドレスタグブロック内のデータが格納されないブロックに対応した論理アドレスにはNULLアドレスを設定することを特徴とする請求項8記載のディスク記憶装置のデータ更新方法。

【請求項10】 上記検査において、各論理アドレスに対するストライプ番号、ストライプ内のブロック番号、有効データのタイムスタンプから構成される変換マップを生成し、ディスク装置の空領域にデータを書き込んだ後、その論理アドレスタグブロックを元に変換マップを修正し、常に有効ブロックを指すように管理することを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項11】 変換マップ作成後、ディスク装置へのアクセスが少ない時間帯に、各ストライプの論理アドレスタグブロックを読み出して変換マップとの比較訂正を行うことを特徴とする請求項10記載のディスク記憶装置のデータ更新方法。

【請求項12】 論理アドレスタグブロックが記録されるディスク装置をストライプによって分散配置し、論理アドレスタグブロックの検査時、ディスク装置の読み出しを並列に行うことを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項13】 ストライプ単位の順次書込みに加え、論理アドレスタグだけを集めた専用タグ領域にもその論理アドレスタグを書込み、論理アドレスタグブロックの検査時にはこの専用タグ領域を順次読み出して調べることが特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項14】 ディスク装置上の記憶領域をストライプ単位に複数のセグメントに分割し、一定期間には1つのセグメントにストライプが書込まれるように制御するとともに、書込むセグメントを切替えるときにはその時点の変換マップと切替え先のセグメント番号をディスク装置に記録し、変換マップ作成時には、ディスク装置のセグメント切替え時の変換マップを見込みディスク装置に記録されたセグメント番号の論理アドレスタグブロックだけを検査することを特徴とする、請求項4記載のディスク記憶装置のデータ更新方法。

【請求項15】 不揮発性メモリ上にセグメント内の各ストライプに対応したビットマップを用意し、書込むセグメントを切替えるときにはこのビットマップをクリアし、ストライプ書込みのときには書込んだストライプに対応したビットをセットし、変換マップ作成時には、ディスク装置のセグメント切替え時の変換マップを見込み、ディスク装置に記録されたセグメント番号の論理アドレスタグの中で、ビットマップがセットされているもののみ検査することを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項16】 タイムスタンプの最小値を進めるた

め、無効ブロックが少ないストライプに対しても定期的に、読み出して有効ブロックだけを前記書込みバッファに移し、対応する論理タグブロック内の論理アドレスと新しいタイムスタンプから論理アドレスタグブロックを生成し、書込みバッファの有効データと生成された論理アドレスタグブロックから構成されるストライプを、読み出したストライプを保持していた領域とは別の空領域に順次書込むことを特徴とする請求項7記載のディスク記憶装置のデータ更新方法。

【請求項17】 タイムスタンプの最小値を進めるため、無効ブロックが少ないストライプに対して定期的に、論理アドレスタグブロックだけを読み出して無効ブロックの論理アドレスをNULLアドレスとした新しいタイムスタンプを付加した論理アドレスタグブロックを生成し、ここで生成された論理アドレスタグブロックを読み出した論理アドレスタグブロックに上書きすることを特徴とする請求項4記載のディスク記憶装置のデータ更新方法。

【請求項18】 変換マップ作成後、ディスク装置上の論理アドレスタグブロックのタイムスタンプと対応する変換マップのタイムスタンプを比較し、無効ブロックを判定することを特徴とする請求項9記載のディスク記憶装置のデータ更新方法。

【請求項19】 N台のディスク装置から構成されるディスク記憶装置において、 $(N-1) \times K$ 個の論理ブロックに相当する容量を持つ書込みバッファを備え、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた $(N-1) \times K$ 個のデータ論理ブロックからK個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えた $N \times K$ 個の論理ブロックをN台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込むことを特徴とするディスク記憶装置のデータ更新方式。

【請求項20】 上記選択した個数として $(N-1) \times K-1$ とし、1個のディスク装置に論理アドレスタグブロックが記録されるようにすることを特徴とする請求項19記載のディスク記憶装置のデータ更新方法。

【請求項21】 上記選択した個数として $(N-1) \times K-2$ とし、1パリティストライプで2個のディスク装置に論理アドレスタグブロックが記録されるように、2つの論理アドレスタグブロックを割り付けることを特徴とする請求項19記載のディスク記憶装置のデータ更新方法。

【請求項22】 ディスク装置に記録された論理アドレ

スタグブロックを検査するのに、パリティストライプ単位の順次書込みに加え、論理アドレスタグを集めた専用タグ領域にもその論理アドレスタグを書込み、この専用タグ領域の書込みデータはパリティで保護しないかわりに、パリティストライプ内の論理アドレスタグが記録されるディスク装置と専用タグ領域の論理アドレスタグが記録されるディスク装置が異なるように専用タグ領域を割り付けることを特徴とする請求項20記載のディスク記憶装置のデータ更新方法。

10 【請求項23】 N台のディスク装置から構成されるディスク記憶装置と、 $N \times K$ (整数) 個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックが $N \times K-1$ 個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、 $N \times K-1$ 個の論理ブロックに前記論理アドレスタグブロックを加えた $N \times K$ 個の論理ブロックを、N台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む制御装置とを具備することを特徴とするディスク記憶制御システム。

20 【請求項24】 書き込みの時間的順序を維持するタイムスタンプが格納される揮発性メモリと、ディスク装置に書き込むべきデータをログ構造化して保持する上記書き込みバッファ、ならびに、書き込みバッファ内の空き領域及び保持されている書き込みデータの論理アドレス情報を保持するバッファ管理情報が格納される不揮発性メモリを具備することを特徴とする請求項23記載のディスク記憶制御システム。

30 【請求項25】 N台のディスク装置から構成されるディスク記憶装置と、 $(N-1) \times K$ 個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた $(N-1) \times K$ 個のデータ論理ブロックからK個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えた $N \times K$ 個の論理ブロックをN台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込制御装置とを具備することを特徴とするディスク記憶制御システム。

40 【請求項26】 パリティを使った冗長性ディスク構成をとるために冗長ディスク装置を付加し、更に、書き込みの時間的順序を維持するタイムスタンプが格納される揮発性メモリと、ディスク装置に書き込むべきデータを

ログ構造化して保持する上記書き込みバッファ、ならびに、書き込みバッファ内の空き領域及び保持されている書き込みデータの論理アドレス情報を保持するバッファ管理情報が格納される不揮発性メモリを具備することを特徴とする請求項25記載のディスク記憶制御システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、特に、RAID (Redundant Array of Inexpensive Disk) 等ディスク 10 アレイの入出力に用いて好適な、ディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システムに関する。

【0002】

【従来の技術】米国特許第5,124,987号、特開平6-214720号、特開平6-266510号に開示されているように、ディスク記憶装置への高速な書き込み方法として、旧データの領域を書き換えるのではなく、更新データを溜めておき、ディスク装置内のあらかじめ用意した別の空領域にまとめて書込む方法が提案されている。

【0003】図17を用い、上述した従来の方法について簡単に説明する。図において、論理ブロックL6, L4, L2, L12, L7, L11を更新する場合を考える。これらの論理ブロックの旧データはディスク装置内のP6, P4, P2, P12, P7, P11に存在する。よって、このP6, P5, P2, P12の内容を更新するのが普通であるが、この方法ではP6, P4, P2, P12, P7, P11のデータはそのままにして、新しい論理ブロックL6, L4, L2, L12, L7, L11のデータをあらかじめ用意した別の空領域であるP51, P52, P53, P54, P55, P56にまとめて 30 書込む。これにより、6回の書き込み操作が3回の書き込み操作に減り、書き込み性能の向上になる。

【0004】尚、図17に示す例では説明を簡単にするため、1ディスクに2ブロックしか書かれていないが、実際には数+ブロックがまとめて書込まれる。また、RAID4、5（いずれもアクセス耐力を向上させる手法であり、RAID4；データの配置単位をビットやバイトのような小さな単位ではなく、セクタやブロックのような大きな単位とし、小容量のデータ読み出し要求に対してディスクを独立動作可とする方式。RAID5；冗 40 長データを専用のパリティディスクに格納するのではなく、各データディスクに巡回的に配置する方式）では1個のストライプ丸ごとの書き換えになるためパリティ維持のためのディスク読み出しも不要になり、書き込み時のオーバーヘッド減少にもなる。

【0005】一方、論理ブロックL6, L5, L2, L12, L7, L11の最新データはディスク装置内のP51, P52, P53, P54, P55, P56に存在するので、間接マップの内容を正しいディスク位置を指すように書き換える。また、論理ブロックのデータを読み出すときには、この間接マップを調べて最 50

新の位置を求めてから読み出すので、旧データを読み出す危険性はない。

【0006】

【発明が解決しようとする課題】上述した従来技術においては、間接マップにより最新データの位置情報を管理していたため、間接マップの障害や誤操作によりそのデータが無くなると、ディスク装置内の全データの喪失になるというデータ安全性の問題があった。また、全論理ブロックに対して間接マップを用意する必要があり大容量、かつ、電源障害に備え間接マップを保持するのに不揮発性メモリが必要であったため、間接マップが非常に高価になるという問題もあった。

【0007】本発明は上記の問題を解決するためになされたものであり、間接マップを原理的に必要としない、安価で高速なディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システムを提供することを目的とする。

【0008】

【課題を解決するための手段】本発明のディスク記憶装置のデータ更新方法は、N台のディスク装置から構成されるディスク記憶装置において、 $N \times K$ （整数）個の論理ブロックに相当する容量を持つ書き込みバッファを有し、この書き込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックのデータ更新を遅延させ、前記書き込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、 $N \times K - 1$ 個の論理ブロックに前記論理アドレスタグブロックを加えた $N \times K$ 個の論理ブロックを、N台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込むことを特徴とする。また、N台のディスク装置から構成されるディスク記憶装置において、 $(N - 1) \times K$ 個の論理ブロックに相当する容量を持つ書き込みバッファを備え、この書き込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書き込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた $(N - 1) \times K$ 個のデータ論理ブロックからK個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えた $N \times K$ 個の論理ブロックをN台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書き込み操作で順次書込むことを特徴とする。

【0009】本発明のディスク記憶制御システムは、N台のディスク装置から構成されるディスク記憶装置と、 $N \times K$ （整数）個の論理ブロックに相当する容量を持つ書き込みバッファと、この書き込みバッファに更新すべきデ

ータの論理ブロックを蓄積し、その蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、 $N \times K - 1$ 個の論理ブロックに前記論理アドレスタグブロックを加えた $N \times K$ 個の論理ブロックを、 N 台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む制御装置とを具備することを特徴とする。また、 N 台のディスク装置から構成されるディスク記憶装置と、 $(N-1) \times K$ 個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた $(N-1) \times K$ 個のデータ論理ブロックから K 個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えた $N \times K$ 個の論理ブロックを N 台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込制御装置とを具備することを特徴とする。

【0010】上記した構成をとることにより、間接マップを原理的に必要としない、安価で高速なディスク記憶装置、ならびにディスク記憶制御システムを構築できる。

【0011】

【発明の実施の形態】図1は本発明を適用して構成したディスク記憶装置の概念構成図である。本発明のディスク記憶制御システムは、制御装置1、ディスク装置2(21, 22, 23, 24)、揮発性メモリ3、不揮発性メモリ4から構成される。揮発性メモリ3には、書込の時間的順序を維持するためのタイムスタンプ5、不揮発性メモリ4には、ディスク装置2に書き込むデータをログ構造化して保持する書込バッファ6、書込バッファ6内の空き領域および保持されている書込データの論理アドレスの情報を保持するバッファ管理情報7が格納されている。制御装置1はこれらタイムスタンプ5、書込バッファ6、バッファ管理情報7を管理し、ディスク装置2への書込みを制御する。

【0012】図2に、不揮発性メモリ4に割り付けられる書込みバッファ6とバッファ管理情報7の関係を示す。制御装置1は外部接続されるホスト機器から要求された書込みデータを、ディスク装置2に対して即書込せずに、ブロック単位に分割して書込みバッファ6に順番に詰めて(ログ形式に)格納していく。この時、書込みデータのホスト機器から見た論理アドレスを、バッファ管理テーブル7の格納したバッファ領域に対応するエン

トリーに保存する。また、そのエントリーにデータが割り当てられたことを示すフラッグ“F”を立てる。よって、このバッファ管理情報(テーブル)7を調べることにより、ホスト機器から受け取った書込みデータを格納すべき次のバッファ領域を決めることが出来る。図2に示す例では、バッファ領域B7まで書込みデータが格納されており、B0, B1, ..., B7の論理アドレスがLA134, LA99, ..., LA678であることを表わしている。

【0013】また、ディスク装置2(21~24)は、それぞれブロックサイズの整数倍(K)であるストライプユニットと呼ぶあらかじめ決められた単位(そのディスク装置の1トラック長に近いサイズが良い。)で書込みを行う。この時、ディスク装置21~24の物理的に同じ位置のストライプユニットは1つのストライプとして、同じタイミングで書込みが行われる。また、実際のディスク装置21~24を合わせて全記憶容量よりも少ない容量のディスク装置としてホスト機器に見せている。具体的には、ホスト機器が最初に記憶容量を問い合わせたとき、その返答として少ない容量を返す。従って、ホスト機器から論理的に読み書きできる記憶領域のほかに余分な記憶領域が確保されることになる。この領域を空領域と呼ぶことにする。

【0014】更に、タイムスタンプ5は、ホストからの書込みデータが実際にディスク装置2に書込まれる時に付加される情報で、ディスク装置2内でのデータ書込み順序を判定するのに用いる。よって、書込みバッファ6のデータがディスク装置2に書込まれる毎にタイムスタンプ5がインクリメントされる。

【0015】図1に示す本発明実施例の動作について図2~図12を参照しながら詳細に説明する。

【0016】まず、書込み動作から説明する。ホスト機器から書込むべきデータとその論理アドレスを受け取った制御装置1は、不揮発性メモリ4上の書込みバッファ6の空領域にブロック単位に分割して詰めて格納する。また、受け取った論理アドレスはブロック毎のアドレスに変換して、バッファ管理テーブル7の対応するエントリーに格納する。なお、既に書込みバッファ6に格納されているデータに対する更新データの場合には、書込みバッファ6の空領域に詰めて格納するのではなく、直接書込みバッファ6内の旧データを変更する。

【0017】ホスト機器からの書込みデータが1ストライプ分に1ブロック少ない数($K \times 4 - 1$)だけ書込みバッファ6に溜まった段階で、制御装置1はそれらのデータをディスク装置2に書込みに行く。この時、最後の書込みブロックとして、書込み管理テーブル7に格納された各ブロックの論理アドレスと揮発性メモリ3上のタイムスタンプ5から論理アドレスタグブロックを作成する。この論理アドレスタグブロック内のアドレスデータとデータブロックの間には、1対1の関係があらかじめ

設定されており、各データブロックの論理アドレスが分かるようになっている。その後、この論理アドレスタグブロックを加えた1ストライプ分のデータを、まとめてディスク装置21~24の空領域に同時に書込む。この様子は図3に示されている。

【0018】また、タイムスタンプ5の値は書込みが完了した段階でインクリメントされる。このように、多数の細かいディスク書込みを1回にまとめられるので、ディスク書込み性能が大きく向上する。

【0019】次に、詰替え処理について説明する。旧データの領域を直接書き換えるのではなく、更新データを溜めておき、ディスク装置2内のあらかじめ用意した別の空領域にまとめて書込む方法では空領域が常に存在することが必須である。そのため、ホスト機器からのディスクアクセスが空いている間に、既に他の領域にデータが書込まれ無効になっているデータを寄せ集めて空領域を作る必要がある。この処理を詰替え処理と呼ぶ。この詰替え処理は無効ブロック判定とストライプ統合の2つのステップからなる。

【0020】無効ブロック判定の例として、図4に示す順番でホスト機器から1ブロックサイズのデータ書込みがある場合を考える。図中のL××はホストから渡される論理アドレス、S××は書込み順番を表わす。本発明実施例では、書込みバッファ6は15ブロックのデータを保持できるので、最初のS1~S15の書込みデータが1つのストライプ(ST1)にまとめられ、タイムスタンプT1が付加されてディスク装置の空領域に書き出される。同様に、S16~S30の書込みデータが別のストライプ(ST2)としてタイムスタンプT2が付加されて別の空領域に書き出される。なお、書き出し毎にタイムスタンプ5はインクリメントされるのでT1<T2の関係がある。

【0021】ここで、図から分かるように論理アドレスL9、L18のデータはタイムスタンプT1のストライプではS5、S2として、タイムスタンプT2のストライプではS19、S21のブロックとして重複して存在する。書込まれた順番を考えると、S19、S21のデータブロックが有効であり、S5、S2のデータは無効と判定されなければならない。しかし、ここで便宜上使った書込み順番S××は実際のディスク上に記録されていない。

【0022】そこで、ストライプ内の論理アドレスタグを使ってこの判定を行う。図4の例における、2つのストライプST1、ST2の論理アドレスタグTG1、TG2の内容は図5の通りである。図から分かるように、2つの論理アドレスタグTG1、TG2に同じ論理アドレスL9、L18のデータが含まれており、ストライプST1のブロックB5、B2とストライプST2のブロックB4、B6のどちらかのデータが無効である。さらに、論理アドレスタグTG1のタイムスタンプT1と論

理アドレスタグTG2のタイムスタンプT2を比較すると、T1<T2の関係にあることから、ストライプST1のブロックB5、B2が無効であることが判定できる。以上説明のように、ディスク装置2内の論理アドレスタグを調べることにより無効なデータブロックを見つけることができる。

【0023】ストライプ統合の例として、図6に示すように2つのストライプST3、ST4を1つのストライプST5に統合する場合を考える。図に示すように、ストライプST3ではB2、B7、B8、B12、B13の5ブロックが有効で他の10ブロックは無効(ハッチング)であるとする。同様に、ストライプST4ではブロックB18、B19、B20、B21、B22、B24、B25、B27、B29の9ブロックが有効で他の6ブロックが無効(ハッチング)であるとする。2つのストライプの有効ブロックは合わせて14ブロックしかないので、この2つのブロックを1つに統合することにより、結果として1つの空領域が作れる。

【0024】ストライプ統合では図6に示すように、2つのストライプST3、ST4を揮発性メモリ3内に読み出し、有効ブロックだけを詰めて書込みバッファ6に移す。それに合わせ、図7に示すように、論理アドレスタグもTG3、TG4から有効ブロックの論理アドレスだけを対応する位置に移し、新しい論理アドレスタグTG5を作りその時点のタイムスタンプを更新する。

【0025】この例では14個の有効ブロックしかなかったので、更にホスト機器から1つの書込みブロックを待ってストライプを完成させてディスク装置2の空領域にまとめて書込む。この場合、ディスク領域は有効に活用されるが、ホスト機器からバーストでディスクアクセスがある場合、書込みを待たすためディスクアクセスを集中させてしまう危険性がある。そこで、最後のデータブロックは空状態のままアクセスが空いている間に書込んでしまうことも可能である。このとき、論理アドレスタグTG5の最後のデータブロックに対する論理アドレスには-1等NULLアドレスを入れることによりデータが入っていないことを表わせるので問題はない。

【0026】次に、このように書かれたデータブロックの読み出し動作について説明する。詰め替え処理の無効ブロック判定を、ディスク装置2内の全ストライプの論理アドレスタグに対して行うことにより、全論理アドレスに対する有効ブロックの物理的位置を検出できる。従って、原理的には、ホスト機器から読み出しブロックの論理アドレスを受け取る度に全ストライプのチェックを行うことにより、読み出すべき物理ブロックを見つけ出すことが出来る。しかし、この方法ではブロック読み出しに膨大な時間が掛ってしまい実用的でない。

【0027】そこで、システム起動時にだけ全ストライプの論理アドレスタグの調査を行ない、揮発性メモリ3上に論理アドレスから物理アドレスへの変換マップを作

る。ホスト機器からの読み出し要求に対してはこの変換マップを使って有効ブロックへのアクセスを行う。これにより、常にアドレスタグの調査をしなくても良く、読み出し時に性能が大きく低下することはない。また、この変換マップは何時でも全ストライプを調査することで再生できるため、従来のように電源障害に備えて不揮発メモリ4に格納する必要もない。

【0028】ここで、変換マップについて図8を用いて説明する。変換マップは図に示すように、各論理アドレスに対するブロックが格納されているストライプ番号S
T#とそのストライプ内のブロック番号B L K #、さらにそのタイムスタンプT S #をテーブル形式で保持している。従って、論理アドレスL 0 ~ L nが与えられ

ば、このテーブルを索引することによりS T #とB L K #から簡単に実際の物理アドレスが求まる。

【0029】また、システム起動時の変換マップの作成は、調査した論理アドレスタグの全論理アドレスについて、テーブルのタイムスタンプより論理アドレスタグのタイムスタンプが大きいときだけ、そのストライプ番号と対応するブロック番号をテーブルに登録する。この調査を全ストライプについて行えば、有効ブロックだけを指す変換マップが出来る。更に、ディスク装置2にストライプを書込む毎に、その論理アドレスタグに対して同様の処理を行うことにより、この変換マップは常に有効なブロックを指す。また、ディスクアクセスが空いているときに、各ストライプの論理アドレスタグと変換マップを比較検査することにより、メモリ障害等でこの変換マップが不正な値になっても検出訂正が可能である。

【0030】以上説明のように、変換マップ作成の主な処理は論理アドレスタグの検査である。故に、大容量ディスク装置のように論理アドレスタグ数が多い場合、電源障害やシステム起動時の変換マップ作成に長時間かかってしまう。特に、図2に示すように、論理アドレスタグブロックが1台のディスク装置24に集中すると、システム起動時にはこのディスクにアクセスが集中し、論理アドレスタグの調査を並列に行うことが出来ない。そこで、図9に示すように、ストライプによって論理アドレスタグが格納されるディスク装置を4台に分散し並列に論理アドレスタグを調査することにより、この変換マップ作成に要する時間を1/4に短縮できる。

【0031】この他、ディスク装置2の記憶領域を複数のセグメントに分割管理することにより、変換マップ作成に必要な論理アドレスタグの検査数を削減できる。図10にセグメント分割方式におけるディスク装置の記憶領域の構成を示す。図に示すように、ディスク装置の記憶領域は、ストライプを単位としてセグメント管理情報(ハッチング)と4つのセグメントに分割される。ここで、セグメントとは書き込みバッファデータの一括書き込みや詰め替え処理のディスク書き込みがある期間集中して行われる単位領域のことである。例えば、セグメント2が

ディスク書き込みの対象である間は、セグメント1、3、4には書き込みに行かないように空領域の選択を制御する。

【0032】また、あるセグメントの空領域が少なくなりディスク書き込みを他のセグメントへ切替えるときにはセグメント管理情報をディスク装置に保存する。セグメント管理情報は図11に示すように、セグメント番号と切替え時変換マップから構成される。セグメント番号とは切替え先のセグメント番号で、切替え時変換マップとはセグメントを切替える時点の揮発性メモリ3上の変換マップの状態である。

【0033】なお、切替え時変換マップはセグメントが切替える度に全て上書きするのではなく、直前のセグメントに書込まれた論理アドレスのエントリだけを書き戻せばよい。従って、前回のセグメント切替え時にタイムスタンプを覚えておき、変換マップのタイムスタンプを比較することにより、直前のセグメントに書込まれた論理アドレスを判定できる。

【0034】このセグメント分割方式では、セグメント切替え時にセグメント管理情報を保存している。よって、セグメント切替え時の変換マップをセグメント管理情報から読み出して、その後でセグメント管理情報のセグメント番号で指されるセグメントの論理アドレスタグだけを検査するだけで、全論理アドレスタグを検査した場合と同じ変換マップが再現できる。従って、この方式により必要な論理アドレスタグの検査数は1セグメント分で良く、この例では変換マップの作成に要する時間を1/4に短縮できる。

【0035】更に、不揮発性メモリ4上にセグメント内の全ストライプに対応したビットマップを用意して、セグメント切替え時にはこのビットマップをクリアし、一括書き込みや詰め替えの書き込み時には書込んだストライプに対応するビットを“1”にセットする。これによりセグメントを切替えてから変更の有ったストライプだけがビットマップが“1”になる。従って、変換マップ作成時にこのビットマップを参照し、変更の有ったストライプの論理アドレスタグだけを検査することで検査数をさらに減らせ、変換マップ作成に要する時間を更に短縮できる。

【0036】通常論理アドレスタグのサイズは512~1024バイトであり、ディスクのシーケンシャルアクセスとランダムアクセスに約50倍の性能差がある。図2に示す方式では論理アドレスタグの情報が各ストライプ毎とびとびに存在するので、変換マップの作成では時間のかかるランダムアクセスを行っていた。そこで、図12に示すように、論理アドレスタグだけを連続して格納する専用タグ領域を(セグメント分割する場合は、各セグメント毎に)用意し、50倍も高速なシーケンシャルアクセスで論理アドレスタグを読み出せるようにする。

【0037】そして、ホスト機器からのデータを一括して書き込みや詰め換えデータの書き込み時には、空領域だけでなく対応する専用タグ領域にも論理アドレスタグを書込むようにする。この方法では図2の方式では1ストライプ当たり4回のディスク書き込みだったのが、専用領域への論理アドレスタグの書き込みのため1回増える。しかし、変換マップ作成が50倍も高速になるので、ディスク装置の立ち上がり時間が問題となるときには非常に有効な手段である。専用タグ領域への書き込み時間を最小にするため、専用タグ領域は図12に示すように対象領域

【0038】最後に、タイムスタンプについて説明する。図1に示すように、タイムスタンプは揮発メモリ3上に記憶されているので、電源障害などにより揮発メモリ3上のタイムスタンプが無くなってしまう。そこで、変換マップと同様にして、システム起動時にだけ全ストライプの論理アドレスタグを調査し、一番大きなタイムスタンプ5の次の値を揮発メモリ3上のタイムスタンプ5にセットする。なお、変換マップ作成の説明で述べた時間短縮手法がそのままタイムスタンプの再生にも適用できる。

【0039】また、タイムスタンプ5はディスク装置に書き込むごとにインクリメントされディスク上の書き込み順序の判定するのに使われる。例として、タイムスタンプ5が24ビットのカウンタで構成される場合で説明する。24ビットカウンタでは、16M回の書き込みでカウンタが一周してゼロに戻ってしまう。そこで、一般的には有効なタイムスタンプの最小値を基準にそれより小さい値は16Mを加えて比較して判定する。この最小値も同様にシステム起動時にだけ全ストライプの論理アドレスタグを調査して求める。

【0040】しかし、この手法が使えるのはタイムスタンプの最大値が最小値を追い越さないこと、つまり、タイムスタンプの最大値と最小値の差が24ビットで表わせる範囲以内であることを前提にしている。よって、タイムスタンプ5が一周前に必ず全ストライプを更新してタイムスタンプ値を新しく更新する必要がある。これには、無効ブロックが少なくとも予め設定した書き込み回数の間更新されなかったストライプを詰替えの対象として選ぶように制御するか、無効ブロックの論理アドレスをNULLアドレスにした、そのストライプの論理アドレスタグだけを書き換える。NULLアドレスを使う方法は論理アドレスタグブロックの書き換えであるので詰替えに比べて非常に軽い処理である。

【0041】尚、上述した実施例では、無効ブロックの判定に2つストライプST1、ST2の論理アドレス

グを相互に比較して判定する方法のみ説明したが、全無効ブロックを調べるには2つのストライプ間の全組み合わせを調べなければならない。しかし、変換マップがあれば論理アドレスタグ内の各論理アドレスについて、有効データを指している変換マップのタイムスタンプとそのストライプのタイムスタンプを比較し、値が小さいブロックを無効ブロックと判定できる。

【0042】図1はデータを複数ディスクに分散するRAID0の構成を示したが、本発明方式は、パリティを使った冗長性ディスク構成(RAID4、5)の場合にも適用できる。図13に本発明を適用して構成したRAID5構成のディスク記憶装置の概念図を示す。図1の構成に冗長用のディスク装置25が追加された構成であり、制御装置1、ディスク装置2(21、22、23、24)、揮発性メモリ3、不揮発性メモリ4、タイムスタンプ5、書込バッファ6、バッファ管理情報7は図1に示す実施例と同じ機能を有する。

【0043】図13に示す実施例の動作につき、図1に示す実施例との差異に着目して説明を行なう。書き込み処理では、ホストからの書き込みデータが1ストライプ分に1ブロック少ない数($K \times 4 - 1$)だけ書込みバッファ6に溜まった段階で、制御装置1はそれらのデータをディスク装置21~25に書き込みに行く。この時、最後の書き込みブロックとして、書込み管理テーブル7に格納された各ブロックの論理アドレスと揮発性メモリ3上のタイムスタンプ5から論理アドレスタグブロックを作成するまでは図1に示す実施例と同じである。

【0044】その後、この論理アドレスタグブロックを加えた1ストライプ分のデータからストライプユニット毎の排他論理演算(XOR)を行い、パリティのストライプユニットを作成する。そして、このパリティ付きのストライプのデータをまとめてディスク装置21~25の空領域に同時に書込む。また、タイムスタンプ5の値は書き込みが完了した段階でインクリメントされる。このように多数の細かいディスク書き込みを1回にまとめられるのに加え、パリティ計算に旧データや旧パリティのブロックを読む必要がないので更にディスクアクセス回数を減らすことができる。なお、ストライプ詰替え処理のディスク書き込みでも同様にしてパリティ付きのストライプを作成してからディスク装置2に書込む。この様子を図14に示す。

【0045】パリティRAID構成では、1台のディスク装置が故障しても、故障したディスクのデータはストライプを構成する他ディスクのデータとパリティのXORを計算することで再現でき、ディスク記憶装置としてのサービスが継続できる。しかし、システム起動時に一台故障していた場合、論理アドレスタグを格納していないディスク装置のデータも読み出して論理アドレスタグを再生してから検査するため、変換マップ作成に時間がかかりシステム起動が完了するまでの時間が大幅に増大

してしまう。

【0046】そこで、図15に示すように、ストライプを構成するデータブロックを1つ減らして2つのディスク装置に同じ論理アドレスタグを書くように制御する。これにより、ディスク装置が1台故障しても変換マップ作成時には残っている方の論理アドレスタグを読み出せるので、システム起動に要する時間の大幅な増大を回避できる。

【0047】また、変換マップ作成の高速化のために専用タグ領域を活用する場合、図16に示されるように、論理アドレスタグが専用タグ領域に格納されるディスク装置と、ストライプに格納されるディスク装置を違うように専用タグ領域の論理アドレスタグの割り付けを制御することにより、ストライプ内の論理アドレスタグは1つでよくなる。

【0048】尚、専用タグ領域へ論理アドレスタグを書く場合もパリティによるディスク障害対策を行うと、従来1回の書き込み増で済んでいたものが、2回の書き込みと2回の読み出しが必要になって一括書き込み時やストライプ詰替え時のディスク書き込みのオーバーヘッドが大きく増大する。従って、この専用タグ領域の情報はパリティで障害対策しない。この情報は変換マップ高速化のためであり、故障したディスク装置の専用タグ領域に格納されていた論理アドレスタグは、ストライプ中のものを(ランダムアクセスで)見れば良いので問題はない。また、ランダムアクセスで検査する論理アドレスタグは1/5だけであるので、変換マップ作成の高速化の効果はある。

【0049】本発明は旧データの領域を書き換えるのではなく、更新データを溜めておきディスク装置内のあらかじめ用意した別の空領域にまとめて書込む方法が有効なあらゆる分野に適用できる。つまり、磁気ディスクだけでなくシーケンシャル書き込みとランダム書き込みで大きく性能が異なる光磁気ディスク等のディスク装置や、小ブロックの更新では2読み出しと2書き込みが必要なパリティによる冗長性を持たせたRAID構成の記憶装置に主に適用できる。

【0050】

【発明の効果】以上説明のように本発明によれば、間接マップを原理的に何時でも再生できるので、電源障害に備えて不揮発性メモリに間接マップに保持する必要はなく、従って非常に安価なディスク記憶装置を構築できる。また、ハードウェア障害等により不揮発性メモリの内容を喪失した場合でも、従来の方法では間接マップが再生できないのでディスク上のデータ全てを失ってしまうのに対し、書き込みバッファに保持されていた最近の書き込みデータだけが失われるだけでほとんどのディスク上データはそのまま残っている。従って、障害に対する耐久性も大きく向上する。更に、電源障害等から回復処理と通常のシステム起動処理は全く同じで済むので、シス

テム終了時や回復時の特別な処理が不要で開発コストを軽減できる。

【0051】また、システム起動時の処理も、複数ディスク装置への論理アドレスタグの分散配置、論理アドレスタグを順次アクセスできる専用タグ領域、記憶領域のセグメント分割管理などにより高速化できるので、システム起動時の待ち時間を実用上問題無い範囲に押さえられる。特に、パリティRAID構成では、論理アドレスタグを2台のディスク装置に記録することにより、1台のディスク装置が故障してもシステム起動の時間が増加しないようにできる。

【図面の簡単な説明】

【図1】本発明の実施例を示すブロック図。

【図2】本発明実施例における書き込みバッファとバッファ管理情報の関係を示すために引用した図。

【図3】本発明実施例におけるディスク装置の空領域に格納される内容を示す図。

【図4】ホスト機器から1ブロックサイズのデータ書き込み順序を示すために引用した図。

【図5】図4の例におけるストライプST1、ST2の論理アドレスタグTG1/TG2の内容を示す図。

【図6】ストライプST3/ST4を1個のストライプST5に統合する例を示す図。

【図7】ストライプ統合において、論理アドレスタグTG3/TG4から論理アドレスタグTG5を作場合の例を示す図。

【図8】本発明実施例において使用される変換マップの構成例を示す図。

【図9】ストライプにより論理アドレスタグが格納されるディスク装置を4台に分散配置した例を示す図。

【図10】セグメント分割におけるディスク装置の記憶領域の割り当てを示す図。

【図11】セグメント管理情報のエントリ構成を示す図。

【図12】論理アドレスタグを連続して格納する専用タグ領域の内容を示す図。

【図13】本発明を適用して構成したRAID5によるディスク装置の実施例を示すブロック図。

【図14】図13に示す実施例の動作概念を示す図。

【図15】2個のディスク装置に同じ論理アドレスタグを書くようにして制御する例を示す図。

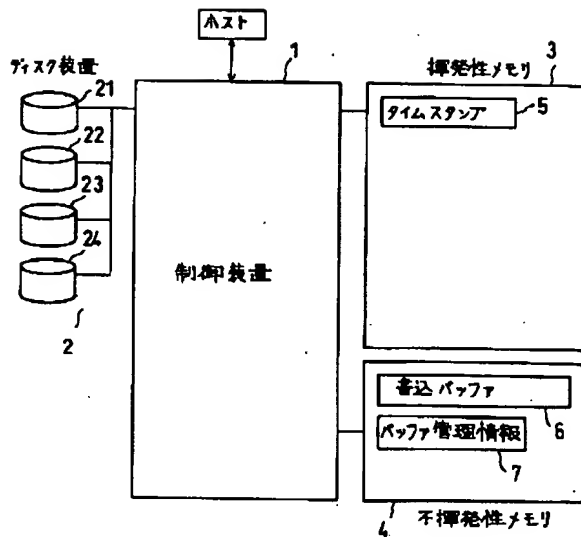
【図16】変換マップ作成の高速化のために、専用タグ領域を割付け使用する場合の例を示す図。

【図17】従来例におけるデータ更新方法を実現するためのシステム構成を示す図。

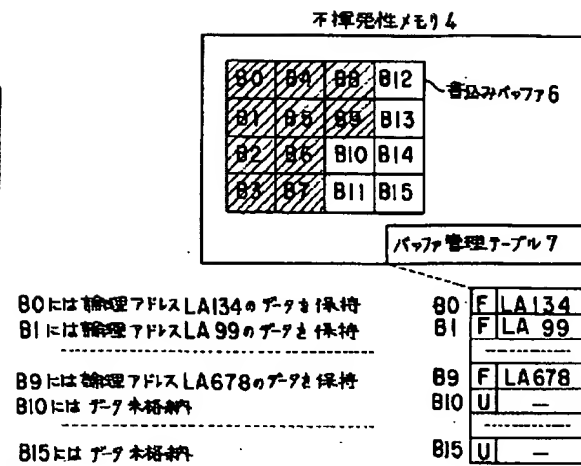
【符号の説明】

1…制御装置、2(21, 22, 23, 24, 25)…ディスク装置、3…揮発性メモリ、4…不揮発性メモリ、5…タイムスタンプ、6…書き込みバッファ、7…バッファ管理情報(テーブル)

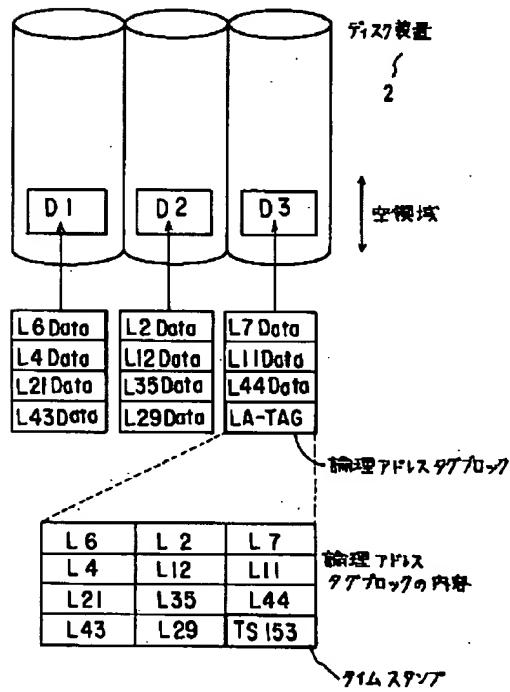
【図1】



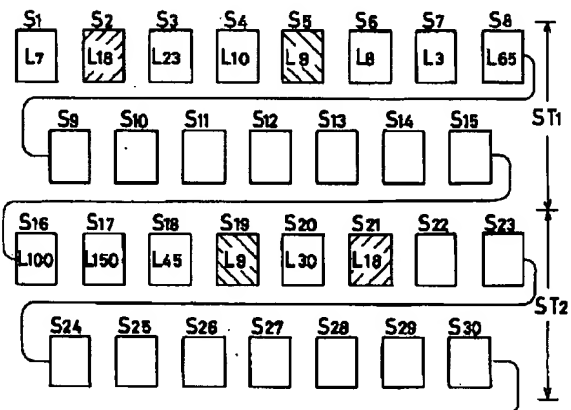
【図2】



【図3】



【図4】



【図8】

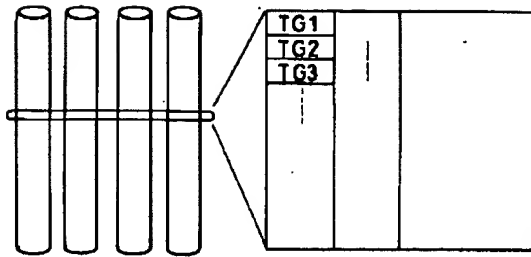
論理アドレス	ST #	BLK #	TS #
L 0			
L 1			
L 2			
⋮			

【図11】

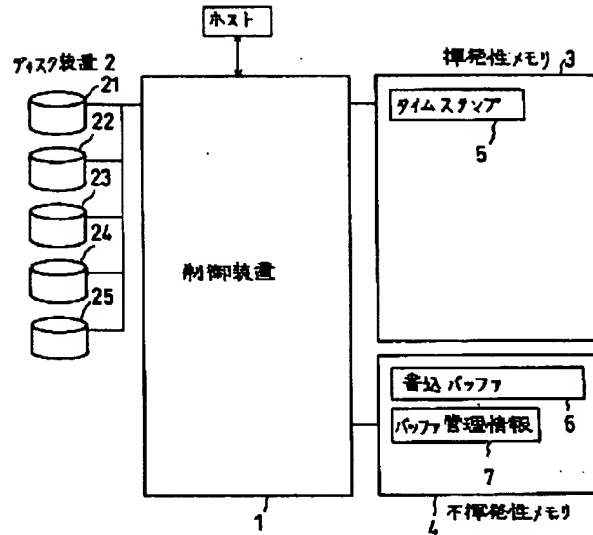
SG #	切替時変換マップ

セグメント管理情報

【図12】

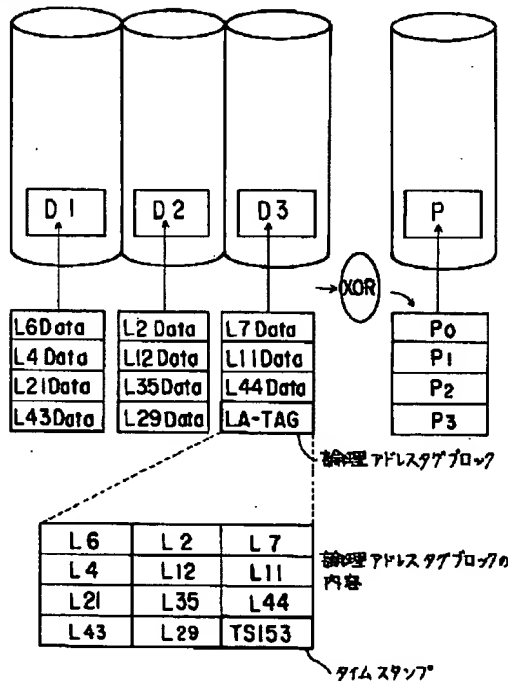


【図13】



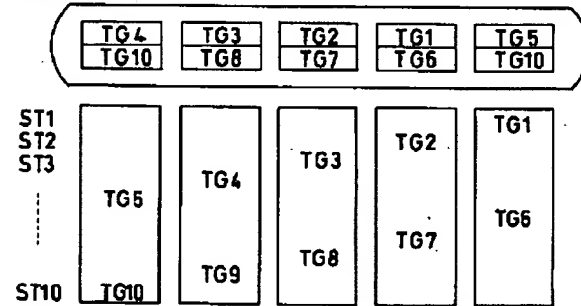
【図14】

ディスク装置 2



【図16】

軸タグ領域



【图 17】

